IDENTIFYING METASTATIC BREAST CANCER USING DEEP TEXTURE REPRESENTATION

Keisuke Fukuta, Daisuke Komura, Tatsuya Harada, Shumpei Ishikawa

The Univercity of Tokyo, Tokyo Medical and Dental University

ABSTRACT

The International Symposium on Biomedical Imaging (ISBI) held a grand challenge to evaluate computational systems for the automated detection of breast cancer metastasis in whole-slide images of histopathological lymph node sections. To solve the challenge, we train patch-based convolutional neural network (CNN). We assume that extracted features from each patch in WSI should be translationally invariant. Therefore, we focus on extracting the texture information as well as spatial one. We use deep texture representation computed with gram-matrix in layer of GoogLeNet[1]. It can learn multiple resolution or size with the same model and can handle large size of image. After training patch-based model, we create tumor probability heatmaps and perform post-processing to make patient-level predictions.

Index Terms— Histopathology, Deep Learning, Texture Representation,

1. INTRODUCTION

Computer-assisted image analysis systems have been developed in a wide range of fields. Especially pathological image classification and detection have become an active research field recently. We develop computational image analysis systems for automatically detecting metastatic breast cancer in digital whole slide images (WSIs) of sentinel lymph node biopsies for Camelyon17.

Most WSI classification or detection methods consist of two parts. Patch-level classification module and post-processing module that aggregates patch-level predictions and outputs the WSI labels. We follow this framework to identify cancer metastases from whole slide images of breast sentinel lymph nodes.

As convolutional neural network (CNN) or deep representation feature using pre-trained CNN has achieved great success in general image analysis, many research applies CNN to pathological images[2]. We assume that the structure of CNN inherently extract spatial information in a patch (i.e. location of each cells), but for classifying patches from WSI, extracted feature is desirably translationally invariant. Therefore, we focus on extracting the texture information as well as spatial one. We use deep texture representation computed with gram-matrix in layer of as pre-trained CNN. We show that this structure outperforms original GoogLeNet and other well-known CNN architectures. In addition, it can learn multiple resolution or size with the same structure while conventional CNN structure cannot.

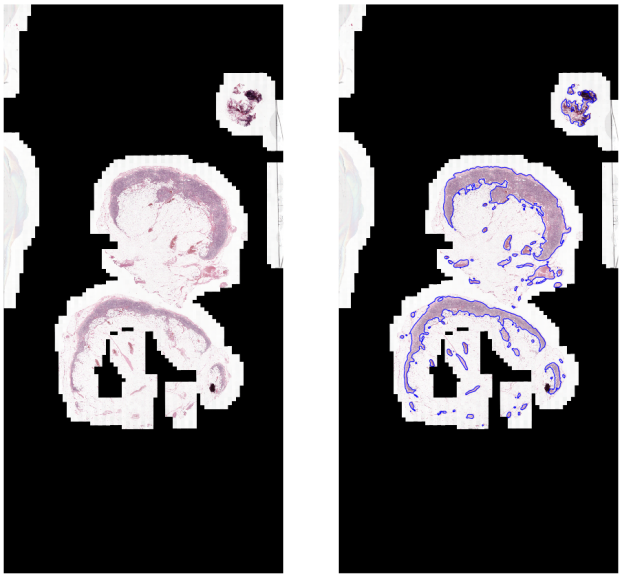

Fig. 1. Visualization of tissue regions detected in image pre-processing (described in Section 2.1). Left: An original whole slide image. Right: Detected tissue regions delineated with the blue contours.

We extract millions of training patches in several resolutions to train our deep neural network model to discriminate tumor patches from normal patches. We then aggregate the patch-level predictions to create tumor probability heatmaps and perform post-processing over these heatmaps to make predictions of slide-level label. After that, we predict the final patient-level label according to pN-staging rule.

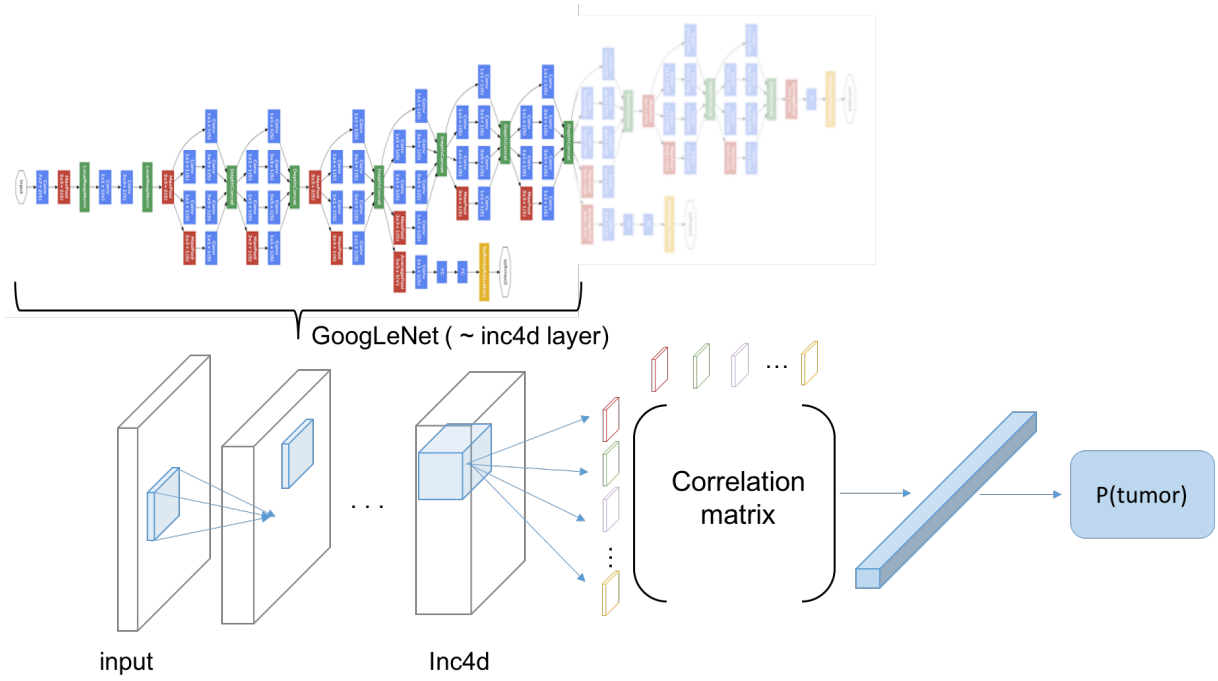

Fig. 2. Architecture of neural network employed in our experiments. The correlation matrix is computed with feature maps of GoogLeNet "inc4d" layer and connected to fully connected layer to output the probability that the patch contains tumor.

2. METHOD

2.1. Image Pre-processing

To reduce computation time and to focus our analysis on regions of the slide likely to contain cancer metastasis, we first identify tissue regions within the WSI and exclude background space. We use WSI in level 6 in OpenSlide. The process flow is as follows:

1. remove black space by threshold in gray scale image.
2. detect and remove white background composed of loose connective tissues and debris in slide by optimal threshold computed with Otsu algorithm [3] in H and S channels in HSV color space. we combine the masks from these two channels with AND operator.
3. apply median filter to remove too small regions using a window size of 8.

An example of identified tissue regions using the pre-process is visualized in Fig.1, where the tissue regions of interest are highlighted with blue contours.

2.2. Patch-based Classification

The patch-based classification model takes as input a patch from WSI and outputs the probability that the patch contains tumor. We use deep neural network using texture representation as a patch-based classification model. In this subsection, we explain the detail of training patch-based classification model.

2.2.1. Extracting Patches

To make a training samples, we first extract randomly 300,000 patches of positive (tumor) and negative (normal) patches from the set of training WSIs. For tumor slide, we only use the slide which has exhaustive lesion-level annotation. If more than half area of the patch contains tumor region, it is labeled positive, otherwise, labeled negative. After selecting the first positive and negative training samples, we train a classification model in a supervised manner to discriminate between these two classes of patches. Next, we apply the classification model to all patches extracted from training WSIs to find hard negative and hard positive samples. [1] Then we make second training samples by replacing easy samples with these hard samples to enrich our training dataset. These technique is called "hard negative mining" often used in object detection tasks. We repeat this process twice and use third training samples for training the final model.

[1] we difine hard samples as misclassified samples with high probability. We set the probability of 0.9.

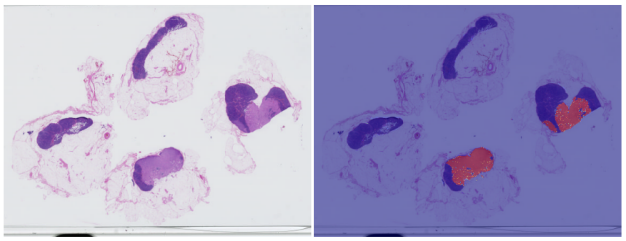

Fig. 3. Visualization of heatmap. Red region indicates high probability to contain tumor. Left: original whole slide image. Right: heatmap overlaid on slide.

2.2.2. Deep Neural Network using texture representation

We use deep texture representation computed with gram-matrix (or correlation matrix) in responses of layers of pre-trained CNN. This manipulation produces an orderless image representation by taking the location-wise outer product of feature maps and aggregating them by averaging. It is closely related to Fisher Vectors, but it has the advantage in the fact that gradients of the model can be easily computed and allows fine-tuning. As showed in Fig.2, We connect a linear classifier and softmax layer on the outputs of the normalized gram-matrix features of the "inc4d" layer outputs in GoogLeNet. The model is similar to recently proposed Bilinear-CNN [4], which use VGG [5] model, instead of GoogLeNet. We tried using VGG based texture representation, but it's unstable and much slower than GoogLeNet based one. Since the model summarizes spatial information, it has no constraint on input size of image size and this is great benefit in patch-based classification, as it can learn multiple resolution or size with the same model.

2.3. Post processing

After training patch-based classification model, we apply this model to all patches from tissue region to create tumor probability heatmaps. On these heatmaps, each pixel has a value p $\in$ [0,1], which indicates probability that the pixel contains tumor. We now consider the post-processing module which takes the heatmaps of WSI of a patient as input and predict pN-stage of the patient. If patch-based classification model is able to completely discriminate between tumor patches from normal patches, we only have to calculate the major axis length of the tumor region to determine slide-level prediction. However, patch-based classification model often makes a mistake and we must determine some threshold. Then, we extract several geometrical features from WSI and train slide-level classification model to automatically determine the threshold. We adopt tree ensemble classifier. Then we aggregate all 5 slide-level predictions to determine the pN-stage of each patient. In training the model, as the final evaluation metrics is not slide-level accuracy but Cohen's kappa score of patient-level pN-stage prediction, we search hyper parameter of tree ensemble which make Cohen's kappa score highest in validation.

3. EXPERIMENT

3.1. Experimental Setup

We used Camelyon16 dataset (400 WSIs) with lesion-level ground truth for training patch-based classification model, and trained post-processing module with Camelyon17 dataset (100 patients and 500 WSIs).

We tried several architectures for patch-based classification model. For training texture model, as it has no constraint on input size, we extracted images in random size chosen from [256, 320, 384, 448, 512] for training samples, other non-texture models in only 256. In all experiments, we used Adam [6] with a learning rate of α = 1e-4. We fixed update weights β_1 = 0.9 and β_2 = 0.999 as recommended in [6]. We trained the model until convergence (about 10 epochs). We used the deep learning framework Chainer [7] version 1.22.0, running on CUDA 8.0 with CuDNN v5.1 on NVIDIA GTX TITAN X GPU with 12GB memory.

To create tumor probability heatmaps, we extracted patches from detected region and evaluated the WSI every 512 image patch with 256 overlapping, or every 256 image patch with no overlapping. We averaged the probabilities of overlapping pixels. For post-processing, we located largest tumor region in the WSI with probability threshold $t = 0.5, 0.75, 0.9$, and then computed the longest axis, area, mean probability and eccentricity of the region. And We computed the number of detected tumor region larger than 1 pixel in the WSI. We used these 15 (5 features $\times$ 3 thresholds) values as input features of tree ensemble model. We adopted Random Forest as post-processing module since it performs better than Gradient Boosting Dicision Tree. We used scikit-learn [8] for training these models and hyper parameters search.

3.2. Results

3.2.1. Patch-level Classification

For classification model, we compared well-known CNN architectures and GoogLeNet based texture model. As shown in Table.1, GoogLeNet based texture model performed the best in terms of AUC. Moreover, texture based model also can take as input images size of 512 $\times$ 512 with the same architecture. The result of 512 $\times$ 512 patches classification is also shown in Table.1.

3.2.2. Patient-level Classification

We considered texture model and conventional non-texture model extracted different features and they might comple-

Table 1. Results of patch-based classification.

Image Size	Model	AUC
256 × 256	**GoogLeNet (texture)**	**0.976**
	GoogLeNet [1]	0.964
	VGG-M (texture)	0.933
	VGG-M [5]	0.952
	ResNet 50 (texture)	0.969
	ResNet 50 [9]	**0.965**
512 × 512	**GoogLeNet (texture)**	**0.966**
	VGG-M (texture)	0.962
	ResNet50 (texture)	0.965

ment each other, and 256 × 256 image and 512 × 512 image might have different information. Therefore, we created 3 tumor probability heatmaps using texture model using 512 × 512 and texture model using 256 × 256, ResNet 50 model. We compared various combinations of these heatmaps for patient-level prediction. When we combine heatmaps, we simply average the probability of the same region. We did 5-fold cross validation for 100 training patients. The results are shown in Table.2.

Table 2. Cohen's kappa scores in cross validation.

	Score
ResNet50	0.711 ± 0.278
Texture 512	0.772 ± 0.098
Texture 256	0.744 ± 0.112
Texture 512 + Texture 256	0.751 ± 0.126
Texture 512 + ResNet50	0.774 ± 0.144
Texture 512 + Texture 256 + ResNet50	**0.810 ± 0.115**

4. DISCUSSION

We developed systems for the automated detection of breast cancer metastasis in whole-slide images of histopathological lymph node sections. We focused on extracting the texture information as well as spatial one and to the best of our knowledge, this is the first study applying deep texture representation to histopatholical image analysis.

Experimental results showed that deep texture model performs better than other conventional models. In addition, as the results indicated that large size of image have much information to detect cancer slide, the fact that the texture model can learn multiple resolution or size with the same structure is a great benefit. Interestingly, when we combined the texture model and conventional model, it showed the best results.

We found detection of ITC with patch-based method is a much challenging task. When we trained patch-based classification model, we define tumor patch as contains tumor region more than its half region. But in some case ITC occupies much small area, and it is difficult to detect the region with high probability. In consequence, our final tree ensenble model learned not to predict ITC in order to achieve high Cohen's kappa score.

We believe that further progress in histopathological image analysis can be made by investigating these texture representation.

5. REFERENCES

[1] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 1–9.

[2] Dayong Wang, Aditya Khosla, Rishab Gargeya, Humayun Irshad, and Andrew H Beck, "Deep learning for identifying metastatic breast cancer," *arXiv preprint arXiv:1606.05718*, 2016.

[3] Nobuyuki Otsu, "A threshold selection method from gray-level histograms," *Automatica*, vol. 11, no. 285-296, pp. 23–27, 1975.

[4] Tsung-Yu Lin and Subhransu Maji, "Visualizing and understanding deep texture representations," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 2791–2799.

[5] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1026–1034.

[6] Diederik Kingma and Jimmy Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[7] Seiya Tokui, Kenta Oono, Shohei Hido, and Justin Clayton, "Chainer: a next-generation open source framework for deep learning," in *Proceedings of workshop on neural information processing systems (NIPS)*, 2015.

[8] Fabian Pedregosa, Gaël Varoquaux, Gramfort, et al., "Scikit-learn: Machine learning in python," *Journal of Machine Learning Research*, vol. 12, no. Oct, pp. 2825–2830, 2011.

[9] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770–778.